PR #35397 完整报告

sgl-project/sglang

Support custom draft worker classes in DSpark

合并时间：2026-08-20 08:48

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35397

执行摘要

- 一句话：DSpark 支持自定义 draft worker 类，默认行为不变。
- 推荐动作：值得快速浏览，作为了解 DSpark 扩展机制的参考。关注点：draft_worker_cls 的类型提示和默认值设计，以及未来若新增测试时需覆盖默认路径回归。

功能与动机

Out-of-tree DSpark 集成可能需要专门的 TpModelWorker，同时希望复用共享的 draft worker 初始化路径。PR body 明确说明动机：Out-of-tree DSpark integrations may need a specialized TpModelWorker while retaining the shared draft-worker initialization path.

实现拆解

实现拆解：

1. 在 python/sglang/srt/speculative/draft_worker_common.py 的 build_draft_tp_worker 函数签名中新增 keyword-only 参数 draft_worker_cls: type[TpModelWorker] = TpModelWorker，默认值保持原有行为；函数内将 TpModelWorker 的实例化替换为 draft_worker_cls(...)，其余初始化逻辑（如 draft_model_build_scope、vocab_size 借用、DraftWorkerBundle 构造）完全不变。
2. 在 python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py 的 DSparkWorkerV2.__init__ 中同样新增 draft_worker_cls 参数并透传给 build_draft_tp_worker，使得 DSpark 的 draft worker 可由外部自定义。
3. 未修改任何测试、配置或文档，因为默认路径未变化，属于纯扩展点添加。

关键文件：

- python/sglang/srt/speculative/draft_worker_common.py（模块 草稿工人；类别 source；类型 core-logic；符号 build_draft_tp_worker）：核心函数 build_draft_tp_worker 新增 draft_worker_cls 参数并调整实例化逻辑，是本次变更的主入口。
- python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py（模块 草稿工人；类别 source；类型 core-logic；符号 DSparkWorkerV2.init）：DSparkWorkerV2.__init__ 透传 draft_worker_cls 给 build_draft_tp_worker，是 DSpark 侧的接入点。

关键符号：build_draft_tp_worker, DSparkWorkerV2.init

关键源码片段

python/sglang/srt/speculative/draft_worker_common.py

核心函数 build_draft_tp_worker 新增 draft_worker_cls 参数并调整实例化逻辑，是本次变更的主入口。

```
# python/sglang/srt/speculative/draft_worker_common.py
# 新增 keyword-only 参数 draft_worker_cls，默认 TpModelWorker，保持向后兼容
def build_draft_tp_worker(
    *,
    server_args: ServerArgs,
    gpu_id: int,
    ps: ParallelState,
    nccl_port: int,
    target_model_config: ModelConfig,
    algo_label: str,
    attention_backend_override: Optional[str] = None,
    draft_worker_cls: type[TpModelWorker] = TpModelWorker,
) -> DraftWorkerBundle:
    # ... 前面的 draft_backend 解析逻辑不变 ...

    with draft_model_build_scope():
        # 使用传入的类来实例化 draft worker，默认还是 TpModelWorker
        draft_worker = draft_worker_cls(
            server_args=server_args,
            gpu_id=gpu_id,
            ps=ps,
            nccl_port=nccl_port,
            is_draft_worker=True,
            context_length=target_model_config.context_len,
            draft_attention_backend=draft_backend,
        )

    draft_model_runner = draft_worker.model_runner
    draft_worker.draft_runner = draft_model_runner
    # ... 后续 vocab_size 借用和 bundle 构造不变 ...
    return DraftWorkerBundle(
        draft_worker=draft_worker,
        draft_model_runner=draft_model_runner,
        draft_model=draft_model_runner.model,
        resolved_attention_backend=draft_backend,
    )
```

python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py

DSparkWorkerV2.__init__ 透传 draft_worker_cls 给 build_draft_tp_worker，是 DSpark 侧的接入点。

```
# python/sglang/srt/speculative/dspark_components/dspark_worker_v2.py
class DSparkWorkerV2(BaseSpecWorker):
```

```
    def __init__(
        self,
        server_args: ServerArgs,
        gpu_id: int,
        ps: ParallelState,
        nccl_port: int,
        target_worker: TpModelWorker,
        draft_worker_cls: type[TpModelWorker] = TpModelWorker, # 新增参数
    ):
        super().__init__()
        # ... 原有初始化逻辑不变 ...

        with self._draft_context():
            bundle = build_draft_tp_worker(
                server_args=server_args,
                gpu_id=gpu_id,
                ps=replace(ps, pp_rank=0),
                nccl_port=nccl_port,
                target_model_config=target_worker.model_runner.model_config,
                algo_label="DSPARK",
                attention_backend_override=(
                    DSV4_DRAFT_ATTENTION_BACKEND if self._draft_is_moe else None
                ),
                draft_worker_cls=draft_worker_cls, # 透传自定义类
            )
        self._draft_worker = bundle.draft_worker
        # ... 后续 draft_model_runner 等字段赋值不变 ...
```

评论区精华

本次 PR 无 review 评论和讨论线程，PR body 中作者提到未运行 GPU 和端到端测试，因为默认路径未变。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：改动为纯参数注入，默认路径不变，不影响现有行为。但存在轻微风险：新增的 draft_worker_cls 参数未加类型校验，若调用方传入不兼容的类，可能导致后续属性访问失败；另外，build_draft_tp_worker 会在 draft_model_build_scope() 上下文中实例化自定义 worker，若自定义类有额外初始化逻辑可能受该 scope 影响。
- 影响：影响范围小：仅影响 DSpark 的初始化路径，默认用户无感知；对需要自定义 draft worker 的 out-of-tree 集成方提供了扩展点，增强扩展性。CRITICAL 风险低，但缺少测试覆盖是主要隐患。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR